

# Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>14</b>
1.1	Ai nên đọc cuốn sách này? . . . . .	20
1.2	Lịch sử các xu hướng trong ngành học sâu . . . . .	23
1.2.1	Những cái tên và vận mệnh đang đổi thay của mạng neuron nhân tạo . . . . .	23
1.2.2	Sự tăng trưởng của kích thước dữ liệu . . . . .	28
1.2.3	Sự tăng trưởng của kích thước mô hình . . . . .	28
1.2.4	Sự tăng trưởng của độ phức tạp, độ chính xác và mức độ ảnh hưởng trong thế giới thực . . . . .	33
<b>I</b>	<b>Cơ bản về toán ứng dụng và học máy</b>	<b>36</b>
<b>2</b>	<b>Đại số tuyến tính</b>	<b>38</b>
2.1	Đại lượng vô hướng, vector, ma trận và tensor . . . . .	38
2.2	Nhân ma trận với vector . . . . .	40
2.3	Ma trận đơn vị và ma trận nghịch đảo . . . . .	42
2.4	Phụ thuộc tuyến tính và span . . . . .	43
2.5	Chuẩn . . . . .	44
2.6	Các ma trận và vector đặc biệt . . . . .	46
2.7	Phân tích riêng . . . . .	47
2.8	Phân tích giá trị suy biến . . . . .	49
2.9	Giả nghịch đảo Moore-Penrose . . . . .	50
2.10	Toán tử vết của ma trận . . . . .	51
2.11	Định thức . . . . .	51
2.12	Ví dụ: Phân tích thành phần chính . . . . .	52
<b>3</b>	<b>Lý thuyết xác suất và lý thuyết thông tin</b>	<b>56</b>
3.1	Tại sao cần xác suất? . . . . .	56
3.2	Biến ngẫu nhiên . . . . .	58
3.3	Phân phối xác suất . . . . .	59
3.3.1	Biến rời rạc và hàm khối xác suất . . . . .	59
3.3.2	Biến liên tục và hàm mật độ xác suất . . . . .	60
3.4	Xác suất biên . . . . .	60
3.5	Xác suất có điều kiện . . . . .	61
3.6	Quy tắc chuỗi của xác suất có điều kiện . . . . .	61
3.7	Độc lập và độc lập có điều kiện . . . . .	61
3.8	Kỳ vọng, phương sai, hiệp phương sai . . . . .	62

3.9	Một số phân phối xác suất thông dụng . . . . .	63
3.9.1	Phân phối Bernoulli . . . . .	63
3.9.2	Phân phối Multinoulli . . . . .	63
3.9.3	Phân phối Gauss . . . . .	64
3.9.4	Phân phối mũ và phân phối Laplace . . . . .	65
3.9.5	Phân phối Dirac và phân phối thực nghiệm . . . . .	66
3.9.6	Hỗn hợp của các phân phối . . . . .	66
3.10	Các tính chất hữu ích của các hàm thông dụng . . . . .	67
3.11	Quy tắc Bayes . . . . .	70
3.12	Chi tiết kỹ thuật về biến số liên tục . . . . .	70
3.13	Lý thuyết thông tin . . . . .	72
3.14	Mô hình xác suất có cấu trúc . . . . .	75
<b>4</b>	<b>Tính toán số</b>	<b>78</b>
4.1	Trần số trên và trần số dưới . . . . .	78
4.2	Tính kém điều hòa . . . . .	79
4.3	Tối ưu dựa trên gradient . . . . .	80
4.3.1	Ma trận Jacobi và ma trận Hesse . . . . .	83
4.4	Tối ưu có ràng buộc . . . . .	87
4.5	Ví dụ: Bình phương cực tiểu tuyến tính . . . . .	89
<b>5</b>	<b>Học máy cơ bản</b>	<b>93</b>
5.1	Các thuật toán học tập . . . . .	94
5.1.1	Tác vụ, $T$ . . . . .	94
5.1.2	Độ đo hiệu năng, $P$ . . . . .	97
5.1.3	Kinh nghiệm $E$ . . . . .	98
5.1.4	Ví dụ: Hồi quy tuyến tính . . . . .	100
5.2	Dung lượng, quá khớp và vị khớp . . . . .	103
5.2.1	Không có bữa trưa nào miễn phí . . . . .	108
5.2.2	Cơ chế kiểm soát . . . . .	109
5.3	Siêu tham số và tập kiểm định . . . . .	111
5.3.1	Kiểm định chéo . . . . .	112
5.4	Ước lượng, độ chệch và phương sai . . . . .	112
5.4.1	Ước lượng điểm . . . . .	112
5.4.2	Độ chệch . . . . .	114
5.4.3	Phương sai và sai số chuẩn . . . . .	116
5.4.4	Đánh đổi giữa độ chệch và phương sai để cực tiểu hóa trung bình bình phương sai số . . . . .	118
5.4.5	Tính nhất quán . . . . .	119
5.5	Ước lượng hợp lý cực đại . . . . .	119
5.5.1	Logarit hàm hợp lý có điều kiện và trung bình bình phương sai số	121
5.5.2	Các tính chất của phương pháp hợp lý cực đại . . . . .	122
5.6	Thống kê Bayes . . . . .	123
5.6.1	Ước lượng hậu nghiệm cực đại (MAP) . . . . .	125
5.7	Các thuật toán học có giám sát . . . . .	126
5.7.1	Học có giám sát dựa trên xác suất . . . . .	126
5.7.2	Máy vector hỗ trợ . . . . .	127
5.7.3	Các thuật toán học có giám sát đơn giản khác . . . . .	129

5.8	Các thuật toán học không giám sát . . . . .	130
5.8.1	Phân tích thành phần chính . . . . .	132
5.8.2	Phân cụm $k$ -means . . . . .	134
5.9	Trượt gradient ngẫu nhiên . . . . .	135
5.10	Xây dựng một thuật toán học máy . . . . .	137
5.11	Những thách thức thúc đẩy sự phát triển của học sâu . . . . .	138
5.11.1	Hiểm họa số chiều lớn . . . . .	138
5.11.2	Tính bất biến cục bộ và cơ chế kiểm soát độ trơn . . . . .	139
5.11.3	Học đa tạp . . . . .	142
<b>II Các kiến trúc học sâu hiện đại</b>		<b>146</b>
<b>6</b>	<b>Mạng lan truyền thuận đa tầng</b>	<b>148</b>
6.1	Ví dụ: Học hàm XOR . . . . .	150
6.2	Học dựa trên gradient . . . . .	155
6.2.1	Hàm chi phí . . . . .	155
6.2.1.1	Học phân phối có điều kiện bằng nguyên tắc hợp lý cực đại	156
6.2.1.2	Học các thống kê có điều kiện . . . . .	157
6.2.2	Các đơn vị đầu ra . . . . .	158
6.2.2.1	Đơn vị tuyến tính cho phân phối đầu ra Gauss . . . . .	158
6.2.2.2	Đơn vị sigmoid cho phân phối đầu ra Bernoulli . . . . .	159
6.2.2.3	Đơn vị softmax cho phân phối đầu ra Multinoulli . . . . .	161
6.2.2.4	Các dạng đầu ra khác . . . . .	163
6.3	Đơn vị ẩn . . . . .	166
6.3.1	Đơn vị tuyến tính hiệu chỉnh (ReLU) và các dạng tổng quát của nó	167
6.3.2	Hàm sigmoid và hàm tanh . . . . .	169
6.3.3	Các đơn vị ẩn khác . . . . .	170
6.4	Thiết kế kiến trúc . . . . .	171
6.4.1	Tính xấp xỉ phổ quát và độ sâu của mạng . . . . .	172
6.4.2	Các yếu tố thiết kế khác cần lưu ý . . . . .	175
6.5	Lan truyền ngược và các thuật toán tính đạo hàm khác . . . . .	176
6.5.1	Đồ thị tính toán . . . . .	176
6.5.2	Quy tắc chuỗi trong phương pháp tính . . . . .	178
6.5.3	Lan truyền ngược bằng cách đệ quy quy tắc chuỗi . . . . .	179
6.5.4	Lan truyền ngược trong mạng MLP kết nối đầy đủ . . . . .	182
6.5.5	Đạo hàm dạng “ký hiệu-ký hiệu” . . . . .	182
6.5.6	Lan truyền ngược tổng quát . . . . .	184
6.5.7	Ví dụ: Lan truyền ngược để huấn luyện MLP . . . . .	187
6.5.8	Một số vấn đề phức tạp . . . . .	189
6.5.9	Phép đạo hàm trong các lĩnh vực khác . . . . .	189
6.5.10	Các đạo hàm bậc cao hơn . . . . .	191
6.6	Vài nét lịch sử . . . . .	191
<b>7</b>	<b>Các cơ chế kiểm soát trong học sâu</b>	<b>195</b>
7.1	Phạt chuẩn của tham số . . . . .	196
7.1.1	Cơ chế phạt chuẩn $L^2$ . . . . .	197
7.1.2	Cơ chế kiểm soát $L^1$ . . . . .	200

7.2	Phạt chuẩn dưới góc nhìn tối ưu có ràng buộc . . . . .	202
7.3	Cơ chế kiểm soát và bài toán không ràng buộc . . . . .	203
7.4	Tăng cường dữ liệu . . . . .	204
7.5	Tính kháng nhiễu . . . . .	206
7.5.1	Thêm nhiễu vào đầu ra . . . . .	207
7.6	Học bán giám sát . . . . .	207
7.7	Học đa nhiệm . . . . .	208
7.8	Kết thúc sớm . . . . .	208
7.9	Trói tham số và dùng chung tham số . . . . .	215
7.9.1	Mạng neuron tích chập . . . . .	216
7.10	Biểu diễn thưa . . . . .	216
7.11	Bagging và các phương pháp hợp thể khác . . . . .	217
7.12	Cơ chế tắt ngẫu nhiên . . . . .	220
7.13	Huấn luyện đối kháng . . . . .	228
7.14	Khoảng cách tiếp tuyến, lan truyền tiếp tuyến và bộ phân loại tiếp tuyến đa tạp . . . . .	230
<b>8</b>	<b>Tối ưu trong huấn luyện các mô hình đa tầng</b>	<b>233</b>
8.1	Tối ưu trong học máy khác tối ưu thuần túy như thế nào? . . . . .	233
8.1.1	Cực tiểu hoá rủi ro thực nghiệm . . . . .	234
8.1.2	Các hàm mất mát thay thế và kết thúc sớm . . . . .	235
8.1.3	Các thuật toán sử dụng lô (batch) và lô nhỏ (minibatch) . . . . .	235
8.2	Những thách thức trong tối ưu mạng neuron . . . . .	239
8.2.1	Tính kém điều hoà . . . . .	240
8.2.2	Cực tiểu cục bộ . . . . .	240
8.2.3	Miền phẳng, điểm yên ngựa và các vùng phẳng khác . . . . .	242
8.2.4	Vách đứng và bùng nổ gradient . . . . .	244
8.2.5	Phụ thuộc dài hạn . . . . .	245
8.2.6	Gradient không chính xác . . . . .	246
8.2.7	Sự liên đới yếu giữa cấu trúc cục bộ và cấu trúc toàn cục . . . . .	246
8.2.8	Giới hạn lý thuyết của tối ưu . . . . .	248
8.3	Các thuật toán cơ bản . . . . .	248
8.3.1	Trượt gradient ngẫu nhiên . . . . .	248
8.3.2	Động lượng . . . . .	250
8.3.3	Động lượng Nesterov . . . . .	253
8.4	Các chiến lược khởi tạo tham số . . . . .	254
8.5	Các thuật toán với tốc độ học tự điều chỉnh . . . . .	258
8.5.1	AdaGrad . . . . .	259
8.5.2	RMSProp . . . . .	259
8.5.3	Adam . . . . .	261
8.5.4	Lựa chọn đúng thuật toán tối ưu . . . . .	261
8.6	Phương pháp xấp xỉ đạo hàm bậc hai . . . . .	262
8.6.1	Phương pháp Newton . . . . .	262
8.6.2	Đạo hàm liên hợp . . . . .	264
8.6.3	BFGS . . . . .	266
8.7	Các chiến thuật tối ưu và các siêu thuật toán . . . . .	267
8.7.1	Chuẩn hóa theo lô . . . . .	267

8.7.2	Trượt theo tọa độ . . . . .	270
8.7.3	Trung bình Polyak . . . . .	271
8.7.4	Tiền huấn luyện có giám sát . . . . .	271
8.7.5	Thiết kế mô hình để hỗ trợ quá trình tối ưu . . . . .	274
8.7.6	Phương pháp liên thông và học theo giáo trình . . . . .	275
<b>9</b>	<b>Các mạng tích chập</b>	<b>278</b>
9.1	Phép tích chập . . . . .	279
9.2	Động lực phát triển . . . . .	281
9.3	Phép gộp . . . . .	285
9.4	Phép tích chập và phép gộp dưới góc nhìn một tiên nghiệm mạnh vô hạn	288
9.5	Các biến thể của hàm tích chập cơ bản . . . . .	290
9.6	Đầu ra có cấu trúc . . . . .	298
9.7	Các kiểu dữ liệu . . . . .	300
9.8	Các thuật toán tích chập hiệu suất cao . . . . .	301
9.9	Đặc trưng ngẫu nhiên hoặc không giám sát . . . . .	302
9.10	Nền tảng thần kinh học của mạng tích chập . . . . .	303
9.11	Mạng tích chập và lịch sử của học sâu . . . . .	309
<b>10</b>	<b>Mô hình hóa chuỗi: Mạng truy hồi và mạng đệ quy</b>	<b>311</b>
10.1	Đồ thị tính toán dàn trải . . . . .	312
10.2	Mạng neuron truy hồi - RNN . . . . .	315
10.2.1	Dạy ép buộc và các mạng có đầu ra truy hồi . . . . .	317
10.2.2	Tính gradient trong mạng truy hồi . . . . .	322
10.2.3	Mạng truy hồi dưới dạng mô hình đồ thị có hướng . . . . .	323
10.2.4	Mô hình hóa chuỗi phụ thuộc ngữ cảnh bằng RNN . . . . .	327
10.3	RNN song hướng . . . . .	329
10.4	Kiến trúc chuỗi-chuỗi dạng “mã hóa-giải mã” . . . . .	330
10.5	Mạng truy hồi sâu . . . . .	332
10.6	Mạng neuron đệ quy . . . . .	334
10.7	Thách thức đến từ phụ thuộc dài hạn . . . . .	336
10.8	Mạng trạng thái vọng hồi . . . . .	339
10.9	Đơn vị rò rỉ và các chiến lược khác cho đa mức thời gian . . . . .	341
10.9.1	Bổ sung kết nối nhảy cóc thời gian . . . . .	341
10.9.2	Đơn vị rò rỉ và phổ của các mức thời gian khác nhau . . . . .	342
10.9.3	Loại bỏ kết nối . . . . .	342
10.10	Bộ nhớ ngắn hạn hướng dài hạn (LSTM) và các RNN sử dụng cổng khác	343
10.10.1	LSTM . . . . .	343
10.10.2	Các RNN sử dụng cổng khác . . . . .	345
10.11	Tối ưu cho phụ thuộc dài hạn . . . . .	346
10.11.1	Siết gradient . . . . .	347
10.11.2	Thúc đẩy luồng thông tin thông qua cơ chế kiểm soát . . . . .	348
10.12	Bộ nhớ tường minh . . . . .	349
<b>11</b>	<b>Phương pháp luận trong thực tế</b>	<b>353</b>
11.1	Độ đo hiệu năng . . . . .	354
11.2	Các mô hình cơ sở mặc định . . . . .	356
11.3	Cân nhắc thu thập thêm dữ liệu . . . . .	357

11.4	Lựa chọn siêu tham số . . . . .	358
11.4.1	Tinh chỉnh thủ công các siêu tham số . . . . .	358
11.4.2	Các thuật toán tối ưu siêu tham số tự động . . . . .	362
11.4.3	Tìm kiếm theo lưới . . . . .	362
11.4.4	Tìm kiếm ngẫu nhiên . . . . .	363
11.4.5	Tối ưu siêu tham số bằng mô hình . . . . .	363
11.5	Chiến lược gỡ lỗi . . . . .	365
11.6	Ví dụ: Nhận dạng số có nhiều chữ số . . . . .	368
<b>12</b>	<b>Ứng dụng</b>	<b>371</b>
12.1	Học sâu ở quy mô lớn . . . . .	371
12.1.1	Triển khai trên CPU tốc độ cao . . . . .	371
12.1.2	Triển khai trên GPU . . . . .	372
12.1.3	Triển khai phân tán quy mô lớn . . . . .	374
12.1.4	Nén mô hình . . . . .	374
12.1.5	Cấu trúc động . . . . .	375
12.1.6	Triển khai các mạng sâu trên phần cứng chuyên dụng . . . . .	377
12.2	Thị giác máy tính . . . . .	378
12.2.1	Tiền xử lý . . . . .	379
12.2.1.1	Chuẩn hoá độ tương phản . . . . .	380
12.2.1.2	Tăng cường dữ liệu . . . . .	383
12.3	Nhận dạng tiếng nói . . . . .	383
12.4	Xử lý ngôn ngữ tự nhiên . . . . .	385
12.4.1	$n$ -gram . . . . .	386
12.4.2	Mô hình ngôn ngữ sử dụng mạng neuron . . . . .	388
12.4.3	Đầu ra có số chiều lớn . . . . .	389
12.4.3.1	Sử dụng danh sách rút gọn . . . . .	389
12.4.3.2	Softmax phân cấp . . . . .	390
12.4.3.3	Lấy mẫu theo độ quan trọng . . . . .	392
12.4.3.4	Ước lượng tương phản nhiễu và hàm mất mát xếp hạng . . . . .	394
12.4.4	Kết hợp mô hình ngôn ngữ neuron với $n$ -gram . . . . .	394
12.4.5	Dịch máy neuron . . . . .	395
12.4.5.1	Sử dụng cơ chế chú ý để liên kết các mảnh dữ liệu . . . . .	397
12.4.6	Vài nét lịch sử . . . . .	397
12.5	Các ứng dụng khác . . . . .	399
12.5.1	Hệ gợi ý . . . . .	400
12.5.1.1	Khám phá và khai thác . . . . .	402
12.5.2	Biểu diễn tri thức, lập luận và hỏi đáp . . . . .	403
12.5.2.1	Tri thức, quan hệ và hỏi đáp . . . . .	403
<b>III</b>	<b>Nghiên cứu học sâu</b>	<b>407</b>
<b>13</b>	<b>Mô hình nhân tử tuyến tính</b>	<b>410</b>
13.1	PCA xác suất và phân tích nhân tử . . . . .	411
13.2	Phân tích thành phần độc lập (ICA) . . . . .	412
13.3	Phân tích đặc trưng chậm . . . . .	414
13.4	Mã hoá thưa . . . . .	416

13.5	Diễn giải PCA dưới góc nhìn đa tạp . . . . .	419
<b>14</b>	<b>Bộ tự mã hóa</b>	<b>422</b>
14.1	Bộ tự mã hóa dưới mức . . . . .	422
14.2	Bộ tự mã hóa có kiểm soát . . . . .	424
14.2.1	Bộ tự mã hóa thừa . . . . .	424
14.2.2	Bộ tự mã hóa khử nhiễu . . . . .	426
14.2.3	Kiểm soát bằng phạt đạo hàm . . . . .	427
14.3	Năng lực biểu diễn, kích thước của tầng và độ sâu . . . . .	427
14.4	Các bộ mã hóa và giải mã ngẫu nhiên . . . . .	428
14.5	Bộ tự mã hóa khử nhiễu . . . . .	429
14.5.1	Ước lượng điểm số . . . . .	430
14.5.1.1	Bối cảnh lịch sử . . . . .	434
14.6	Học đa tạp với bộ tự mã hóa . . . . .	434
14.7	Bộ tự mã hóa co ngắn . . . . .	437
14.8	Phân tích thừa dự đoán . . . . .	442
14.9	Ứng dụng của các bộ tự mã hóa . . . . .	442
<b>15</b>	<b>Học biểu diễn</b>	<b>444</b>
15.1	Huấn luyện trước không giám sát theo tầng . . . . .	445
15.1.1	Khi nào và tại sao huấn luyện trước không giám sát mang lại hiệu quả? . . . . .	447
15.2	Học chuyển giao và thích ứng miền . . . . .	452
15.3	Bóc tách các yếu tố nhân quả thông qua học bán giám sát . . . . .	455
15.4	Biểu diễn phân tán . . . . .	460
15.5	Lợi ích mang lại tăng theo hàm mũ của độ sâu . . . . .	465
15.6	Cung cấp manh mối cho quá trình học để khám phá ra các nguyên nhân cơ bản . . . . .	467
<b>16</b>	<b>Các mô hình xác suất có cấu trúc trong học sâu</b>	<b>470</b>
16.1	Thách thức trong việc mô hình hoá phi cấu trúc . . . . .	471
16.2	Mô tả cấu trúc của mô hình bằng đồ thị . . . . .	474
16.2.1	Mô hình có hướng . . . . .	474
16.2.2	Mô hình vô hướng . . . . .	476
16.2.3	Phân hàm . . . . .	478
16.2.4	Mô hình năng lượng . . . . .	479
16.2.5	Tách biệt và tách biệt-D . . . . .	480
16.2.6	Chuyển đổi giữa đồ thị vô hướng và có hướng . . . . .	483
16.2.7	Đồ thị nhân tử . . . . .	485
16.3	Lấy mẫu từ mô hình đồ thị . . . . .	486
16.4	Ưu điểm của mô hình hóa có cấu trúc . . . . .	488
16.5	Học các quan hệ phụ thuộc . . . . .	488
16.6	Suy diễn và suy diễn xấp xỉ . . . . .	489
16.7	Sử dụng học sâu trong mô hình xác suất có cấu trúc . . . . .	490
16.7.1	Ví dụ: Máy Boltzmann hẹp . . . . .	492
<b>17</b>	<b>Phương pháp Monte Carlo</b>	<b>494</b>
17.1	Lấy mẫu và các phương pháp Monte Carlo . . . . .	494

17.1.1	Vì sao cần lấy mẫu? . . . . .	494
17.1.2	Cơ bản về phương pháp lấy mẫu Monte Carlo . . . . .	495
17.2	Lấy mẫu theo độ quan trọng . . . . .	496
17.3	Phương pháp Monte Carlo sử dụng chuỗi Markov . . . . .	498
17.4	Lấy mẫu Gibbs . . . . .	501
17.5	Thách thức trong việc pha trộn các mode tách biệt . . . . .	502
17.5.1	Điều hoà để pha trộn các mode . . . . .	504
17.5.2	Độ sâu có thể hỗ trợ pha trộn . . . . .	505
<b>18</b>	<b>Đối mặt với phân hàm</b>	<b>507</b>
18.1	Gradient của logarit hàm hợp lý . . . . .	507
18.2	Hợp lý cực đại ngẫu nhiên và phân kỳ tương phản . . . . .	509
18.3	Hàm hợp lý giả . . . . .	515
18.4	Khớp điểm số và khớp tỷ lệ . . . . .	517
18.5	Khớp điểm số khử nhiễu . . . . .	519
18.6	Ước lượng tương phản nhiễu . . . . .	519
18.7	Ước lượng phân hàm . . . . .	521
18.7.1	Lấy mẫu theo độ quan trọng bằng phép ủ . . . . .	523
18.7.2	Lấy mẫu bắc cầu . . . . .	526
<b>19</b>	<b>Suy diễn thống kê xấp xỉ</b>	<b>528</b>
19.1	Suy diễn thông qua tối ưu hóa . . . . .	528
19.2	Cực đại hóa kỳ vọng . . . . .	530
19.3	Suy diễn MAP và mã hóa thừa . . . . .	532
19.4	Suy diễn biến phân và học biến phân . . . . .	533
19.4.1	Biến tiềm ẩn rời rạc . . . . .	535
19.4.2	Phương pháp tính biến phân . . . . .	539
19.4.3	Biến tiềm ẩn liên tục . . . . .	542
19.4.4	Tương tác giữa học và suy diễn . . . . .	543
19.5	Học cách suy diễn xấp xỉ . . . . .	544
19.5.1	Thức-ngủ . . . . .	544
19.5.2	Các dạng thức khác của học suy diễn . . . . .	545
<b>20</b>	<b>Mô hình sinh đa tầng</b>	<b>546</b>
20.1	Máy Boltzmann . . . . .	546
20.2	Máy Boltzmann hẹp . . . . .	548
20.2.1	Phân phối có điều kiện . . . . .	548
20.2.2	Huấn luyện máy Boltzmann hẹp . . . . .	550
20.3	Mạng niềm tin đa tầng . . . . .	551
20.4	Máy Boltzmann đa tầng . . . . .	553
20.4.1	Các tính chất đáng chú ý . . . . .	555
20.4.2	Suy diễn mean-field trong DBM . . . . .	556
20.4.3	Học tham số trong DBM . . . . .	557
20.4.4	Huấn luyện trước theo tầng . . . . .	558
20.4.5	Huấn luyện DBM đồng thời . . . . .	561
20.5	Máy Boltzmann cho dữ liệu có giá trị thực . . . . .	564
20.5.1	RBM dạng Gauss-Bernoulli . . . . .	564
20.5.2	Mô hình vô hướng của hiệp phương sai có điều kiện . . . . .	566



20.6	Máy Boltzmann tích chập . . . . .	569
20.7	Máy Boltzmann cho đầu ra có cấu trúc hoặc đầu ra tuần tự . . . . .	571
20.8	Các máy Boltzmann khác . . . . .	572
20.9	Lan truyền ngược qua phép toán ngẫu nhiên . . . . .	573
20.9.1	Lan truyền ngược qua phép toán ngẫu nhiên rời rạc . . . . .	574
20.10	Mạng sinh mẫu có hướng . . . . .	577
20.10.1	Mạng niềm tin sigmoid . . . . .	577
20.10.2	Mạng sinh mẫu khả vi . . . . .	578
20.10.3	Bộ tự mã hoá biến phân . . . . .	580
20.10.4	Mạng đối kháng sinh mẫu . . . . .	583
20.10.5	Mạng sinh mẫu khớp moment . . . . .	585
20.10.6	Mạng sinh mẫu tích chập . . . . .	586
20.10.7	Mạng tự hồi quy . . . . .	587
20.10.8	Mạng tự hồi quy tuyến tính . . . . .	587
20.10.9	Mạng tự hồi quy neuron . . . . .	588
20.10.10	NADE . . . . .	589
20.11	Lấy mẫu từ bộ tự mã hoá . . . . .	591
20.11.1	Chuỗi Markov tương ứng với bộ tự mã hoá khử nhiễu bất kỳ . . . . .	592
20.11.2	Ghim và lấy mẫu có điều kiện . . . . .	593
20.11.3	Thủ tục huấn luyện đi lùi . . . . .	593
20.12	Mạng sinh mẫu ngẫu nhiên . . . . .	594
20.12.1	Mạng sinh mẫu ngẫu nhiên theo hướng phân loại . . . . .	594
20.13	Các chiến lược sinh mẫu khác . . . . .	595
20.14	Đánh giá mô hình sinh mẫu . . . . .	596
20.15	Kết luận . . . . .	598
	<b>Chú giải thuật ngữ</b>	<b>599</b>
	<b>Bibliography</b>	<b>600</b>

# Ký hiệu

Chúng tôi cung cấp một mô tả ngắn gọn về các ký hiệu sẽ được dùng thống nhất trong cuốn sách này. Chúng tôi sẽ mô tả hầu hết các khái niệm toán học dưới đây trong các chương 2 - 4.

## Số và Mảng (array)

$a$	Đại lượng vô hướng (kiểu nguyên hoặc kiểu thực)
$\mathbf{a}$	Vector
$\mathbf{A}$	Ma trận
$\mathbf{A}$	Tensor
$\mathbf{I}_n$	Ma trận đơn vị $n$ hàng $n$ cột
$\mathbf{I}$	Ma trận đơn vị với số chiều tùy thuộc ngữ cảnh
$\mathbf{e}^{(i)}$	Vector cơ sở chính tắc $[0, \dots, 0, 1, 0, \dots, 0]$ với 1 ở vị trí $i$
$\text{diag}(\mathbf{a})$	Ma trận đường chéo với các phần tử của đường chéo chính là $\mathbf{a}$
$a$	Biến ngẫu nhiên dạng vô hướng
$\mathbf{a}$	Biến ngẫu nhiên dạng vector
$\mathbf{A}$	Biến ngẫu nhiên dạng ma trận

## Tập hợp và đồ thị

$\mathbb{A}$	Tập hợp
$\mathbb{R}$	Tập hợp số thực
$\{0, 1\}$	Tập hợp bao gồm 0 và 1
$\{0, 1, \dots, n\}$	Tập hợp bao gồm tất cả các số nguyên giữa 0 và $n$
$[a, b]$	Khoảng số thực bao gồm cả $a$ và $b$
$(a, b)$	Khoảng số thực không bao gồm $a$ nhưng bao gồm $b$
$\mathbb{A} \setminus \mathbb{B}$	Tập hợp trừ, nghĩa là tập hợp bao gồm các phần tử thuộc $\mathbb{A}$ mà không thuộc $\mathbb{B}$
$\mathcal{G}$	Đồ thị
$\text{Pa}_{\mathcal{G}}(x_i)$	Nút cha của $x_i$ trong đồ thị $\mathcal{G}$

**Đánh chỉ số**

- $a_i$  Phần tử thứ  $i$  của vector  $\mathbf{a}$ , được đánh chỉ số từ 1
- $a_{-i}$  Tất cả các phần tử của vector  $\mathbf{a}$  trừ phần tử thứ  $i$
- $A_{i,j}$  Phần tử ở hàng thứ  $i$  cột thứ  $j$  của ma trận  $\mathbf{A}$
- $\mathbf{A}_{i,:}$  Hàng thứ  $i$  của ma trận  $\mathbf{A}$
- $\mathbf{A}_{:,i}$  Cột thứ  $i$  của ma trận  $\mathbf{A}$
- $A_{i,j,k}$  Phần tử ở vị trí  $(i, j, k)$  của một tensor 3-D  $\mathbf{A}$
- $\mathbf{A}_{::,i}$  Lát cắt 2-D của một tensor 3-D
- $a_i$  Phần tử thứ  $i$  của một vector ngẫu nhiên  $\mathbf{a}$

**Toán tử đại số tuyến tính**

- $\mathbf{A}^\top$  Chuyển vị của ma trận  $\mathbf{A}$
- $\mathbf{A}^+$  Giải nghịch đảo Moore-Penrose của  $\mathbf{A}$
- $\mathbf{A} \odot \mathbf{B}$  Tích theo từng phần tử (Hadamard) của  $\mathbf{A}$  và  $\mathbf{B}$
- $\det(\mathbf{A})$  Định thức của  $\mathbf{A}$

**Phương pháp tính (calculus)**

- $\frac{dy}{dx}$  Đạo hàm của  $y$  theo  $x$
- $\frac{\partial y}{\partial x}$  Đạo hàm riêng của  $y$  theo  $x$
- $\nabla_{\mathbf{x}} y$  Gradient của  $y$  theo  $\mathbf{x}$
- $\nabla_{\mathbf{X}} y$  Ma trận đạo hàm của  $y$  theo  $\mathbf{X}$
- $\nabla_{\mathbf{x}} y$  Tensor chứa đạo hàm của  $y$  theo  $\mathbf{X}$
- $\frac{\partial f}{\partial \mathbf{x}}$  Ma trận Jacob  $\mathbf{J} \in \mathbb{R}^{m \times n}$  của  $f : \mathbb{R}^n \leftarrow \mathbb{R}^m$
- $\nabla_{\mathbf{x}}^2 f(\mathbf{x})$  hoặc  $\mathbf{H}(f)(\mathbf{x})$  Ma trận Hesse của  $f$  tại điểm đầu vào  $\mathbf{x}$
- $\int f(\mathbf{x}) d(\mathbf{x})$  Tích phân xác định trên toàn bộ miền của  $\mathbf{x}$
- $\int_{\mathbb{S}} f(\mathbf{x}) d(\mathbf{x})$  Tích phân xác định của  $\mathbf{x}$  trên tập hợp  $\mathbb{S}$

### Lý thuyết xác suất và lý thuyết thông tin

$a \perp b$	Hai biến ngẫu nhiên $a$ và $b$ độc lập
$a \perp b \mid c$	Hai biến ngẫu nhiên $a$ và $b$ độc lập có điều kiện khi biết $c$
$P(a)$	Phân phối xác suất của một biến rời rạc
$p(a)$	Phân phối xác suất của một biến liên tục hoặc một biến có kiểu chưa xác định
$a \sim P$	Biến ngẫu nhiên $a$ có phân phối $P$
$\mathbb{E}_{x \sim P}[f(x)]$ hoặc $\mathbb{E}f(x)$	Kỳ vọng của $f(x)$ theo $P(x)$
$\text{Var}(f(x))$	Phương sai của $f(x)$ theo $P(x)$
$\text{Cov}(f(x), g(x))$	Hiệp phương sai của $f(x)$ và $g(x)$ theo $P(x)$
$H(x)$	Entropy Shannon của biến ngẫu nhiên $x$
$D_{\text{KL}}(P \parallel Q)$	Độ phân kỳ Kullback-Leibler của $P$ và $Q$
$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Phân phối Gaussian của $\mathbf{x}$ với trung bình $\boldsymbol{\mu}$ và hiệp phương sai $\boldsymbol{\Sigma}$

### Hàm số

$f : \mathbb{A} \leftarrow \mathbb{B}$	Hàm số $f$ với miền xác định $\mathbb{A}$ và miền giá trị $\mathbb{B}$
$f \circ g$	Phép hợp của hai hàm $f$ và $g$
$f(\mathbf{x}; \boldsymbol{\theta})$	Hàm số của $\mathbf{x}$ được tham số hoá bởi $\boldsymbol{\theta}$ . (Đôi khi chúng tôi viết là $f(\mathbf{x})$ và bỏ qua đối số $\boldsymbol{\theta}$ để tinh giản ký hiệu)
$\log x$	Logarit tự nhiên của $x$
$\sigma(x)$	Hàm logit sigmoid, $\frac{1}{1+\exp(-x)}$
$\zeta(x)$	Hàm softplus, $\log(1 + \exp(x))$
$\ \mathbf{x}\ _p$	Chuẩn $L^p$ của $\mathbf{x}$
$\ \mathbf{x}\ $	Chuẩn $L^2$ của $\mathbf{x}$
$x^+$	Phần dương của $x$ , tức là $\max(0, x)$
$\mathbf{1}_{\text{condition}}$	trả về 1 nếu thoả mãn điều kiện, ngược lại trả về 0

Đôi khi, chúng ta sử dụng hàm  $f$  có đối số vô hướng, nhưng lại áp dụng nó cho một vector, một ma trận hoặc một tensor:  $f(\mathbf{x})$ ,  $f(\mathbf{X})$  hoặc  $f(\mathbf{X})$ . Điều này biểu thị việc áp dụng hàm  $f$  theo từng phần tử của mảng. Ví dụ if  $\mathbf{C} = \sigma(\mathbf{X})$ , then  $C_{i,j,k} = \sigma(X_{i,j,k})$  đối với tất cả các giá trị hợp lệ của  $i, j$  và  $k$ .

**Tập dữ liệu và phân phối**

$P_{\text{data}}$	Phân phối sinh dữ liệu
$\hat{p}_{\text{data}}$	Phân phối thực nghiệm, định nghĩa bởi tập huấn luyện
$\mathbf{x}^{(i)}$	Mẫu thứ $i$ (đầu vào) của một tập dữ liệu
$y^{(i)}$ hoặc $\mathbf{y}^{(i)}$	Nhãn tương ứng với $\mathbf{x}^{(i)}$ trong bài toán học có giám sát
$\mathbf{X}$	Ma trận $m \times n$ , với mẫu đầu vào $\mathbf{x}^{(i)}$ nằm ở hàng $\mathbf{X}_i$ .

# Chương 1

## Giới thiệu

*ND: Những phần có ghi ND là chú giải thêm của người dịch*

Từ xa xưa, con người đã có ước mơ tạo ra những cỗ máy có khả năng suy nghĩ. Khoa khát này đã có từ thời Hy Lạp cổ đại. Những hình tượng thần thoại như Pygmalion, Daedalus, Hephaestus là những nhà phát minh vĩ đại, và Galatea, Talos, Pandora có thể được xem là các thực thể sống nhân tạo ([OM04]; [Spa96]; [Tan97])

Ngay từ khi những chiếc máy tính đầu tiên có khả năng lập trình được tạo ra, người ta đã tự hỏi xem liệu chúng có thể trở nên thông minh hay không, cho đến khi một cỗ máy thông minh như kỳ vọng ra đời sau đó một thế kỷ ([Lov42]). Ngày nay, trí tuệ nhân tạo (*artificial intelligence* - AI) đã trở thành một lĩnh vực với vô vàn ứng dụng thực tiễn và trở thành chủ đề thu hút rất nhiều các đề tài nghiên cứu trên toàn thế giới. Chúng ta sử dụng phần mềm thông minh để tự động hóa các công việc chân tay, nhận dạng hình ảnh, âm thanh, chẩn đoán y học và hỗ trợ nghiên cứu khoa học cơ bản.

Trong những ngày đầu của trí tuệ nhân tạo, ngành này đã nhanh chóng giải quyết những vấn đề tuy phức tạp đối với con người, nhưng lại tương đối đơn giản đối với máy tính - những vấn đề có thể diễn tả được bằng một danh sách những quy luật dưới dạng ngôn ngữ toán học chuẩn tắc. Tuy nhiên, thách thức thực sự đối với trí tuệ nhân tạo là giải quyết những tác vụ dễ thực hiện với con người nhưng khó diễn tả một cách chuẩn tắc - những vấn đề mà con người chúng ta xử lý một cách rất tự nhiên bằng trực giác, chẳng hạn như nhận dạng tiếng nói hoặc nhận dạng khuôn mặt.

Nội dung cuốn sách này xoay quanh một giải pháp cho những vấn đề ấy. Giải pháp này cho phép máy tính tự học từ những kinh nghiệm thu được và hiểu hơn về thế giới thông qua một hệ phân cấp các khái niệm, trong đó mỗi *khái niệm* (concept) được định nghĩa theo mối quan hệ của nó với những khái niệm đơn giản hơn. Bằng cách để máy tính tự động thu thập tri thức từ kinh nghiệm, cách tiếp cận này giúp giảm bớt gánh nặng cho con người trong việc mô tả các tri thức cần thiết cho máy tính một cách tường minh. *Hệ phân cấp khái niệm* cho phép máy tính học những khái niệm phức tạp từ những khái niệm đơn giản hơn. Nếu chúng ta vẽ một đồ thị miêu tả hệ phân cấp khái niệm này, thì đồ thị đó sẽ có rất nhiều lớp, và rất *sâu*. Vì vậy, ta gọi cách tiếp cận này là *học sâu* (deep learning).

Rất nhiều thành công ban đầu của AI ra đời trong phòng thí nghiệm, một môi trường không đòi hỏi máy tính phải có nhiều hiểu biết về thế giới bên ngoài. Ví dụ, hệ thống chơi cờ vua *Deep Blue* của công ty IBM đã đánh bại nhà vô địch thế giới Garry Kasparov vào năm 1997 ([Hsu02]). Cờ vua dĩ nhiên là một môi trường đơn giản, chỉ bao gồm 64 ô vuông và 32 quân cờ, trong đó mỗi quân cờ chỉ có thể di chuyển theo những quy tắc

xác định. Dù sáng tạo ra một chiến thuật chơi cờ hiệu quả là một thành tựu lớn, nhưng thách thức không nằm ở việc dạy cho máy tính hiểu về các quân cờ và cách di chuyển của chúng. Luật chơi cờ vua được biểu diễn đầy đủ bằng một số quy tắc và có thể lập trình một cách dễ dàng.

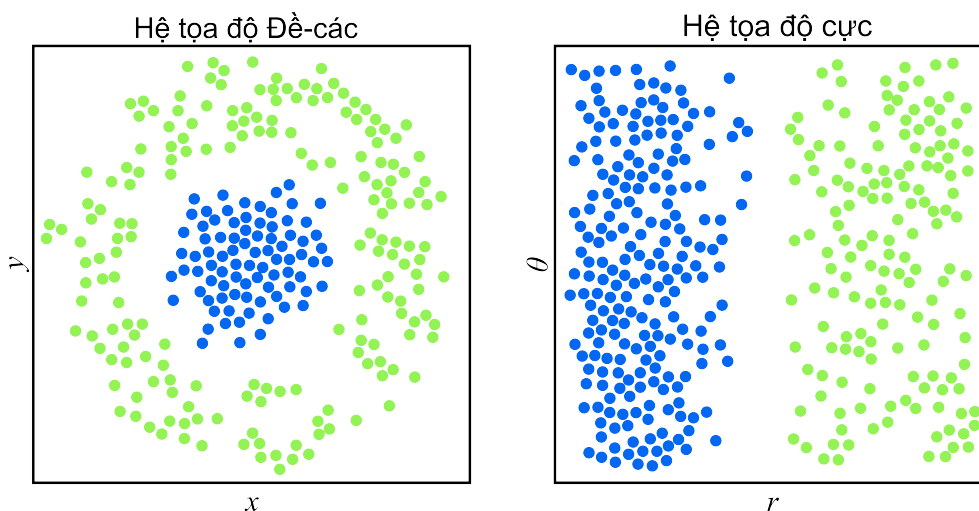
Trở trêu thay, những tác vụ trừu tượng và chuẩn tắc khiến con người gặp trở ngại lại là những tác vụ dễ nhất với máy tính. Máy tính từ lâu đã có khả năng đánh bại người chơi cờ vua giỏi nhất, nhưng mới chỉ theo kịp trình độ của người bình thường trong việc nhận dạng vật thể và tiếng nói trong thời gian gần đây. Cuộc sống hàng ngày của một con người đòi hỏi một lượng tri thức khổng lồ về thế giới. Phần lớn những tri thức này mang tính chủ quan và phụ thuộc vào trực giác của mỗi người, nên rất khó để có thể diễn đạt chúng một cách tường minh. Máy tính cần nắm bắt được những tri thức như vậy về thế giới để có thể trở nên thông minh. Đưa những tri thức không tường minh vào trong một chiếc máy tính là một trong những thách thức chính trong ngành trí tuệ nhân tạo.

Một số dự án AI cố gắng gắn cứng tri thức về thế giới thông qua ngôn ngữ chuẩn tắc, để máy tính có thể tư duy về các chỉ lệnh trong ngôn ngữ chuẩn tắc thông qua các quy tắc suy luận logic. Phương pháp này được gọi là tiếp cận thông qua *cơ sở tri thức* (knowledge base). Tuy nhiên, không một dự án nào trong số đó đạt được thành công lớn. Một trong số những dự án nổi tiếng nhất là Cyc ([LG89]). Cyc là một công cụ suy luận và là một cơ sở dữ liệu gồm các chỉ lệnh được viết bằng ngôn ngữ CycL. Những chỉ lệnh này được đội ngũ nhân viên giám sát của Cyc đưa vào. Đó là một quá trình đầy gian nan. Người ta gặp khó khăn trong việc tạo ra những quy tắc tỉ mỉ, rất phức tạp để cố gắng miêu tả thế giới một cách chính xác nhất. Ví dụ, Cyc không hiểu được câu chuyện về một người tên Fred đang cạo râu vào buổi sáng ([Lin92]). Công cụ suy luận của Cyc đã gặp phải một chi tiết mâu thuẫn trong câu chuyện: nó biết rằng con người không có những bộ phận điện tử trên cơ thể, nhưng vì lúc đó Fred đang cầm một chiếc máy cạo râu, cho nên nó cho rằng thực thể mang tên “FredWhileShaving” có chứa những bộ phận điện tử. Cho nên Cyc đặt ra câu hỏi liệu Fred có còn là con người khi anh ta đang cạo râu hay không?

Những khó khăn mà các mô hình tri thức gắn cứng gặp phải cho thấy một hệ thống trí tuệ nhân tạo thực thụ cần có khả năng tự thu thập tri thức, bằng việc trích xuất các *mô thức* (pattern) từ dữ liệu thô. Khả năng này được gọi là *học máy* (machine learning). Sự xuất hiện của *học máy* cho phép máy tính giải quyết các vấn đề cần đến tri thức về thế giới thực và đưa ra quyết định chủ quan. Một thuật toán *học máy* đơn giản như *hồi quy logit* (logistic regression) có thể chẩn đoán và khuyến nghị có nên thực hiện phẫu thuật để hỗ trợ sản phụ khi sinh hay không ([Mor+90]). Một thuật toán khác, gọi là *giả luận Bayes* (naive Bayes), có thể phân biệt email thông thường và email rác.

Hiệu năng của những thuật toán học máy đơn giản này phụ thuộc nhiều vào *biểu diễn* (representation) của dữ liệu đầu vào. Ví dụ, khi sử dụng hồi quy logit để chẩn đoán khả năng cần phẫu thuật khi sinh, hệ thống AI không kiểm tra bệnh nhân một cách trực tiếp. Thay vào đó, các bác sĩ nạp một vài thông tin liên quan vào hệ thống, ví dụ như có tồn tại sẹo tử cung hay không. Mỗi mảnh thông tin liên quan đến bệnh nhân được gọi là một *đặc trưng* (feature). Hồi quy logit sẽ học mối tương quan giữa mỗi đặc trưng với nhiều kết quả đầu ra. Tuy nhiên, thuật toán này không thể quyết định việc các đặc trưng được định nghĩa thế nào. Nếu hồi quy logit được cung cấp một bản chụp MRI của bệnh nhân, thay vì báo cáo chi tiết của bác sĩ, nó sẽ không thể đưa ra dự đoán chính xác. Những điểm ảnh đơn lẻ trong bản chụp MRI không có nhiều tương quan với các biến chứng có thể xảy ra trong khi thực hiện phẫu thuật.

Sự phụ thuộc vào biểu diễn là hiện tượng phổ biến trong khoa học máy tính và cuộc sống hàng ngày. Trong khoa học máy tính, những thao tác như tìm kiếm trong một tập hợp dữ liệu có thể diễn ra nhanh hơn theo cấp số nhân nếu tập hợp đó được tổ chức có cấu trúc và được lập chỉ mục một cách thông minh. Đa phần chúng ta có thể dễ dàng thực hiện các phép toán số học trên hệ chữ số Ả Rập nhưng lại mất nhiều thời gian khi thực hiện trên hệ La Mã. Vậy nên không có gì ngạc nhiên khi việc lựa chọn cách biểu diễn có ảnh hưởng lớn đến hiệu suất của các thuật toán *học máy*. Hình 1.1 minh họa một ví dụ trực quan.



Hình 1.1: Ví dụ về sự khác biệt trong cách biểu diễn: giả sử chúng ta muốn phân loại hai dạng dữ liệu bằng cách vẽ một đường thẳng phân tách chúng trên *biểu đồ phân tán* (scatterplot). Trong hình bên trái, dữ liệu được biểu diễn theo hệ tọa độ Descartes (Đề-Các), và đường thẳng phân tách dữ liệu không tồn tại. Trong hình bên phải, ta biểu thị dữ liệu dưới hệ tọa độ cực và ta chỉ cần vẽ một đường thẳng dọc là có thể phân đôi được hai tập điểm. Hình được vẽ với sự giúp đỡ của David Warde-Farley.

Nhiều tác vụ trí tuệ nhân tạo có thể được giải quyết bằng cách thiết kế những đặc trưng phù hợp, rồi cung cấp bộ đặc trưng đó cho một thuật toán học máy đơn giản. Ví dụ, một đặc trưng hữu ích cho tác vụ nhận dạng người nói từ một đoạn âm thanh là kích cỡ thanh quản ước lượng của người đó. Đặc trưng này cho phép ta nhận biết được người nói là nam giới, nữ giới hay một đứa trẻ.

Tuy nhiên, trong nhiều tác vụ, rất khó để biết được những đặc trưng nào cần trích xuất. Ví dụ, chúng ta muốn viết một chương trình phát hiện xe hơi trong ảnh. Biết rằng xe hơi có bánh, nên ta cho sự xuất hiện của bánh xe là một đặc trưng. Tuy nhiên, rất khó để miêu tả chính xác hình dạng của một chiếc bánh xe thông qua giá trị của các điểm ảnh. Một chiếc bánh xe có hình dạng rất đơn giản, nhưng ảnh chụp của nó có thể trở nên phức tạp hơn nhiều do bóng đổ lên bánh xe, ánh nắng chói lóa ở những bộ phận kim loại của bánh xe, tấm chắn bùn của xe hoặc một vật thể nào đó che khuất vài bộ phận của bánh xe, và nhiều thứ khác nữa.

Một giải pháp cho vấn đề này là sử dụng học máy để không chỉ tìm được phép ánh xạ từ biểu diễn tới đầu ra, mà còn khám phá ra phép ánh xạ từ biểu diễn tới một biểu diễn khác. Hướng tiếp cận này được gọi là *học biểu diễn* (representation learning). Những biểu diễn học được thường hiệu quả hơn biểu diễn được thiết kế thủ công. Chúng cũng cho phép các hệ thống AI thích nghi nhanh với các tác vụ mới với rất ít sự can thiệp từ con



người. Một thuật toán học biểu diễn có thể tìm ra bộ đặc trưng phù hợp cho những tác vụ đơn giản trong vài phút, còn với những tác vụ phức tạp, nó cũng chỉ cần vài giờ tới vài tháng. Việc thiết kế thủ công các đặc trưng cho một tác vụ phức tạp yêu cầu nhiều thời gian và nỗ lực của con người, có thể khiến cả một cộng đồng đông đảo các nhà nghiên cứu tốn hàng thập kỷ để thực hiện.

Một ví dụ kinh điển của thuật toán học biểu diễn là *bộ tự mã hóa* (auto-encoder). Một *bộ tự mã hóa* là sự kết hợp của một *bộ mã hóa* (encoder), có chức năng biến đổi dữ liệu đầu vào thành một biểu diễn khác, và một *bộ giải mã* (decoder), có chức năng đưa biểu diễn mới trở về dạng ban đầu. Bộ tự mã hóa được huấn luyện để giữ lại nhiều thông tin nhất có thể khi dữ liệu đầu vào đi qua bộ mã hóa và bộ giải mã, nhưng chúng cũng được huấn luyện để các biểu diễn mới này có nhiều tính chất thú vị. Tồn tại nhiều loại bộ tự mã hóa khác nhau được thiết kế để đạt được những đặc tính mong muốn.

Khi thiết kế đặc trưng hay thiết kế thuật toán để học đặc trưng, mục tiêu của chúng ta là tách rời các *biến tố* (factors of variation) giúp giải thích dữ liệu quan sát được. Chữ **tố** ở đây, viết gọn của **nhân tố** (factor) ám chỉ những nguồn ảnh hưởng riêng lẻ. Những nhân tố như vậy thường là các đại lượng ta không quan sát được. Chúng có thể tồn tại dưới dạng các đối tượng không thể quan sát hay các lực không thể đo đạc trong thế giới vật chất, nhưng có tác động đến những đại lượng ta quan sát thấy. Chúng còn có thể tồn tại dưới dạng các thành tố trong tâm trí con người, giúp đưa ra cách giải thích đơn giản hơn hay suy diễn những nguyên nhân đằng sau dữ liệu thu thập được. Ta có thể coi chúng như những khái niệm hay những dạng trừu tượng giúp giải thích sự đa dạng trong dữ liệu. Ví dụ, khi phân tích một đoạn ghi âm lời nói, biến tố bao gồm tuổi, giới tính, chất giọng và những câu từ của người nói. Khi phân tích một bức ảnh chụp xe hơi, biến tố bao gồm vị trí, màu sắc, góc quan sát chiếc xe và độ sáng của ánh nắng mặt trời.

Một thách thức đối với nhiều ứng dụng AI trong thực tế đó là có nhiều biến tố ảnh hưởng tới mọi mảnh dữ liệu mà ta quan sát được. Những điểm ảnh trong bức ảnh chụp chiếc xe hơi màu đỏ có thể có màu gần với màu đen nếu ảnh được chụp vào ban đêm. Hình dạng cái bóng của chiếc xe tùy thuộc vào góc quan sát. Hầu hết các ứng dụng đòi hỏi việc *bóc tách* được các biến tố và loại bỏ những nhân tố không quan trọng khác.

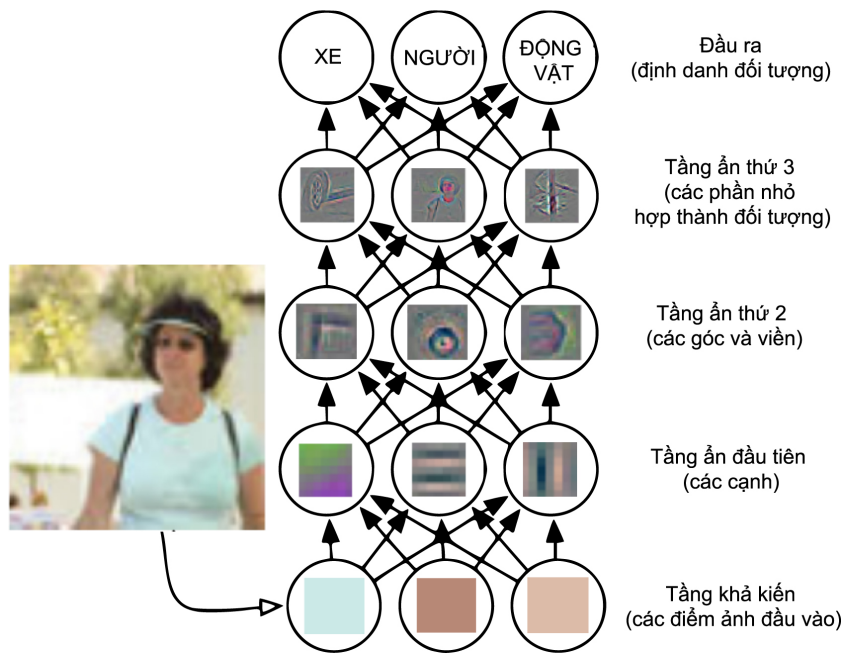
Đĩ nhiên, việc trích xuất các đặc trưng mức cao, trừu tượng từ dữ liệu thô có thể rất khó. Đa số những biến tố, chẳng hạn như chất giọng người nói, chỉ có thể được xác định khi AI có trình độ hiểu biết tinh tế, gần ngang trình độ của con người. Khi việc đạt được biểu diễn phù hợp có độ khó tương đương với giải quyết bài toán ban đầu, gần như học biểu diễn không giúp được gì nhiều cho chúng ta.

*Học sâu* (deep learning) giải quyết vấn đề trọng tâm này của học biểu diễn bằng cách đưa ra biểu diễn mới dựa trên những biểu diễn đơn giản hơn. Học sâu cho phép máy tính xây dựng các khái niệm phức tạp trên cơ sở là những khái niệm đơn giản hơn. Hình 1.2 cho thấy một hệ thống học sâu có thể biểu diễn khái niệm về một bức ảnh chụp con người bằng cách kết hợp các khái niệm đơn giản hơn, như các góc và viền, những khái niệm này lại được xác định từ khái niệm đơn giản hơn là các cạnh.

Ví dụ điển hình cho một mô hình học sâu là *mạng lan truyền thuận đa tầng* (feed forward deep network), hay thường được gọi là *mạng perceptron đa tầng* (multilayer perceptron - MLP). Một MLP thực chất là một hàm toán học ánh xạ dữ liệu đầu vào tới các giá trị đầu ra. Hàm này được hợp thành từ những hàm đơn giản hơn. Chúng ta có thể coi mỗi lần áp dụng một hàm toán học khác nhau là một lần đưa ra cách biểu diễn mới cho dữ liệu đầu vào.

Ý tưởng về nhu cầu học biểu diễn đúng cho dữ liệu đã cho ta một góc nhìn về học

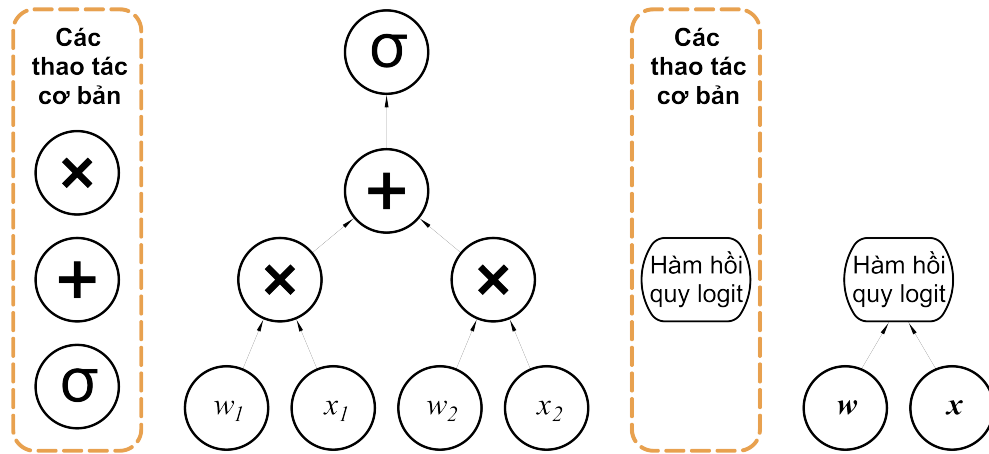
sâu. Một góc nhìn khác về học sâu đó là độ sâu của nó cho phép máy tính học ra một chương trình nhiều công đoạn. Mỗi tầng biểu diễn có thể được xem như trạng thái của bộ nhớ máy tính sau khi thực hiện một loạt các câu lệnh song song. Các mạng với độ sâu lớn hơn có thể thực hiện nhiều chuỗi lệnh hơn. Các chuỗi lệnh có công năng rất lớn, bởi những câu lệnh sau có thể sử dụng lại kết quả của những câu lệnh trước. Dưới góc nhìn này của học sâu, không nhất thiết rằng tất cả thông tin trong các *đơn vị kích hoạt* (activation) của một tầng đều mã hóa những biến tố giải thích dữ liệu đầu vào. Biểu diễn cũng lưu trữ các thông tin trạng thái giúp thực thi một chương trình có thể hiểu được dữ liệu đầu vào. Thông tin trạng thái này có thể tương đương với bộ đếm hay con trỏ trong một chương trình máy tính truyền thống. Dù không làm việc với nội dung cụ thể của đầu vào, nhưng nó giúp mô hình tổ chức quá trình xử lý.



Hình 1.2: Minh họa một mô hình học sâu. Thật khó để máy tính hiểu được ý nghĩa của các dữ liệu đầu vào ở dạng thô, chẳng hạn như bức ảnh này, nó được biểu diễn bằng một tập hợp các giá trị điểm ảnh. Ánh xạ từ tập các điểm ảnh đến định danh của vật thể là một hàm cực kỳ phức tạp. Trực tiếp học và đánh giá hàm ánh xạ này có vẻ như là nhiệm vụ bất khả thi. Học sâu giải quyết khó khăn này bằng cách chia nhỏ hàm ánh xạ phức tạp cần tìm thành một chuỗi các hàm ánh xạ đơn giản lồng vào nhau, mỗi ánh xạ được mô tả bởi một tầng khác nhau của mô hình. Đầu vào được đặt tại *tầng khả kiến* (visible layer), nó được đặt tên như vậy là bởi các biến số ở tầng này có thể quan sát được. Tiếp theo là một chuỗi các *tầng ẩn* (hidden layer) giúp trích xuất các đặc trưng có mức độ trừu tượng tăng dần. Chúng được gọi là “tầng ẩn” bởi giá trị ở những tầng này không có sẵn trong dữ liệu; thay vào đó, mô hình phải tự xác định những khái niệm nào là hữu ích trong việc lý giải mối liên hệ trong dữ liệu quan sát được. Những hình ảnh ở đây là hiển thị của các đặc trưng được biểu diễn ở mỗi tầng. Từ các điểm ảnh cho trước, tầng thứ nhất có thể dễ dàng nhận ra các cạnh biên, bằng cách so sánh độ sáng giữa các điểm ảnh lân cận với nhau. Từ các cạnh biên đã được mô tả bởi tầng ẩn thứ nhất, tầng ẩn thứ hai có thể dễ dàng tìm ra các góc và viền, có thể được nhận diện như tập hợp của các cạnh biên. Từ các góc và viền được mô tả bởi tầng ẩn thứ hai, tầng ẩn thứ ba có thể phát hiện ra toàn bộ thành phần của các vật thể cụ thể, bằng cách tìm các tập hợp đường bao và góc cụ thể. Cuối cùng, các mô tả dưới dạng các bộ phận của vật thể có thể được dùng để nhận diện vật thể trong ảnh. Những hình ảnh trên được tái hiện với sự cho phép của [ZF14].

## CHƯƠNG 1. GIỚI THIỆU

Có hai cách thức chính để đánh giá độ sâu của một mô hình. Một là dựa trên chuỗi các lệnh cần thực hiện khi thực thi toàn bộ kiến trúc. Ta có thể xem nó như độ dài của đường đi dài nhất trong lưu đồ mô tả cách tính đầu ra của mỗi mô hình ứng với với đầu vào tương ứng. Cũng giống như việc hai chương trình máy tính tương đương sẽ có độ dài khác nhau phụ thuộc vào ngôn ngữ được sử dụng để viết chúng, cùng một hàm số có thể có lưu đồ với các độ sâu khác nhau phụ thuộc vào hàm chúng ta được phép sử dụng trong các bước đơn lẻ của lưu đồ. Hình 1.3 minh họa việc lựa chọn ngôn ngữ có thể dẫn tới độ sâu khác nhau cho cùng một kiến trúc như thế nào.



Hình 1.3: Minh họa một biểu đồ tính toán ánh xạ một đầu vào tới đầu ra, với mỗi node thực hiện một phép toán. Độ sâu là chiều dài của đường đi dài nhất từ đầu vào tới đầu ra, nhưng nó phụ thuộc vào định nghĩa thế nào là một bước tính toán. Biểu đồ trên minh họa quá trình tính toán của mô hình hồi quy logit,  $\sigma(\mathbf{w}^T \mathbf{x})$ , trong đó  $\sigma$  là hàm sigmoid. Nếu ta sử dụng phép cộng, nhân và hàm sigmoid là các phép toán cơ bản trong ngôn ngữ lập trình, thì mô hình này có độ sâu là ba. Nếu ta coi hồi quy logit như là một phép toán cơ bản thì mô hình này chỉ có độ sâu là một.

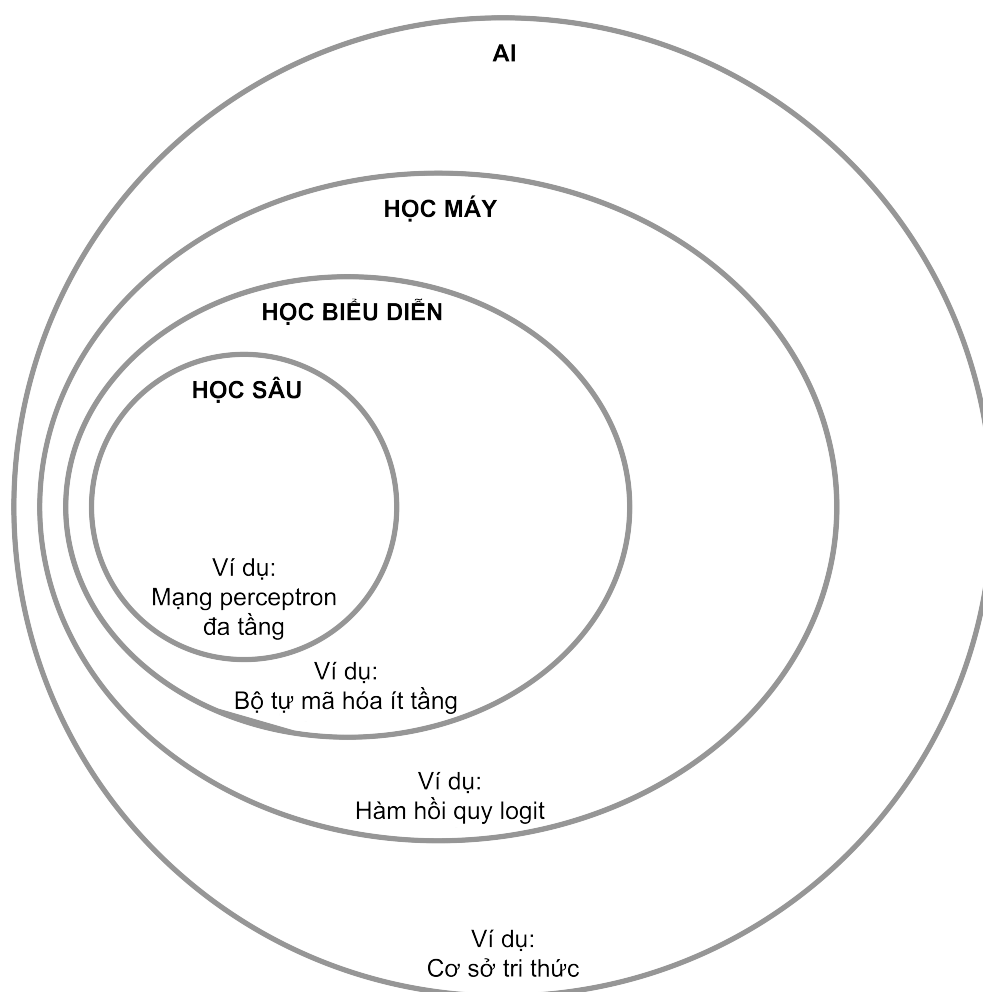
Cách thức xác định độ sâu thứ hai, sử dụng bởi các *mô hình đồ thị xác suất sâu* (deep graphical model), định nghĩa chiều sâu của một mô hình không phải là chiều sâu của lưu đồ tính toán, mà là của đồ thị mô tả liên hệ giữa các khái niệm với nhau. Trong trường hợp này, độ sâu của lưu đồ mô tả quá trình tính toán biểu diễn của mỗi khái niệm có thể lớn hơn nhiều so với đồ thị của chính khái niệm đó, bởi cách hiểu của hệ thống về các khái niệm đơn giản có thể được tinh chỉnh khi chúng ta có thêm thông tin về các khái niệm phức tạp hơn. Ví dụ, một hệ thống AI quan sát một hình ảnh của một khuôn mặt với một mắt trong bóng tối ban đầu sẽ chỉ nhìn thấy một mắt. Sau đó phát hiện rằng có một khuôn mặt, hệ thống có thể suy diễn rằng có thể tồn tại con mắt thứ hai trong ảnh. Trong trường hợp này, đồ thị của khái niệm chỉ bao gồm hai tầng, một tầng cho mắt và một tầng cho khuôn mặt, nhưng đồ thị tính toán lại bao gồm  $2n$  tầng nếu chúng ta thực hiện tinh chỉnh ước lượng về mỗi khái niệm  $n$  lần khi biết các khái niệm khác.

Rất khó để xác định góc nhìn nào về độ sâu là phù hợp hơn - độ sâu của đồ thị tính toán hay độ sâu của mô hình đồ thị xác suất - và bởi vì mỗi người có cách chọn tập các thành phần cơ bản khác nhau để xây dựng đồ thị cho riêng mình, nên ta không có một giá trị độ sâu chính xác duy nhất cho một kiến trúc, tương tự với việc không có một giá trị duy nhất nào cho độ dài của một chương trình máy tính. Và chúng ta cũng không có quy ước về một mô hình như thế nào thì được xem là “sâu”. Tuy vậy, chúng ta có thể coi học sâu là lĩnh vực nghiên cứu những mô hình bao gồm cả sự kết hợp một số lượng lớn

hơn các hàm hoặc các khái niệm học được, so với các mô hình học máy truyền thống.

Tóm lại, học sâu, chủ đề của cuốn sách này, là một hướng tiếp cận dành cho các bài toán AI. Cụ thể, nó là một dạng của học máy, một kỹ thuật cho phép hệ thống máy tính tự học từ trải nghiệm và dữ liệu. Chúng tôi dám chắc rằng học máy là hướng tiếp cận khả thi duy nhất để xây dựng các hệ thống AI có thể vận hành trong thế giới thực phức tạp. Học sâu là một dạng cụ thể của học máy, sở hữu sức mạnh và sự linh hoạt tuyệt vời thông qua việc học cách biểu diễn thế giới như một hệ phân cấp các khái niệm, trong đó mỗi khái niệm được định nghĩa từ những khái niệm đơn giản hơn, và mỗi biểu diễn được tính toán từ những biểu diễn kém trừu tượng hơn.

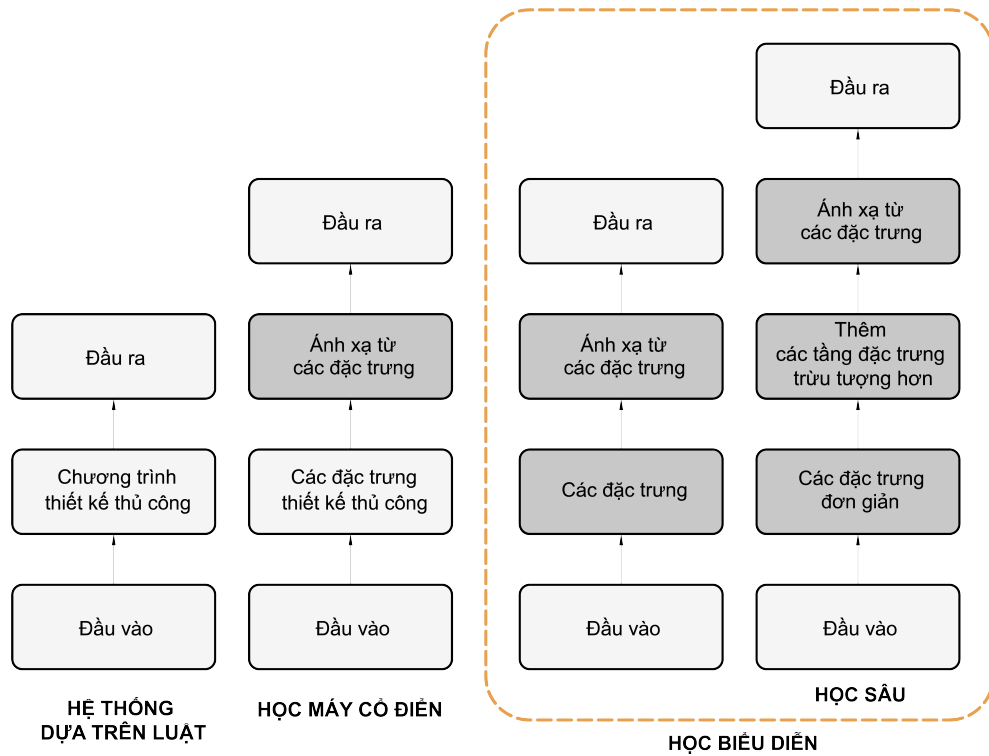
Hình 1.4 thể hiện mối liên hệ về phạm vi của các lĩnh vực khác nhau trong AI. Hình 1.5 mô tả một giản đồ về cách hoạt động của mỗi loại.



Hình 1.4: Biểu đồ Venn cho thấy học sâu là một dạng học biểu diễn, và học biểu diễn lại là một dạng học máy, một ngành được sử dụng trong nhiều (nhưng không phải toàn bộ) hướng tiếp cận AI. Mỗi phần của biểu đồ Venn bao gồm một ví dụ của một công nghệ AI.

### 1.1 Ai nên đọc cuốn sách này?

Cuốn sách có thể hữu ích với rất nhiều đối tượng độc giả, nhưng chúng tôi hướng tới hai đối tượng chính. Thứ nhất là sinh viên các trường đại học (hoặc sau đại học) đang



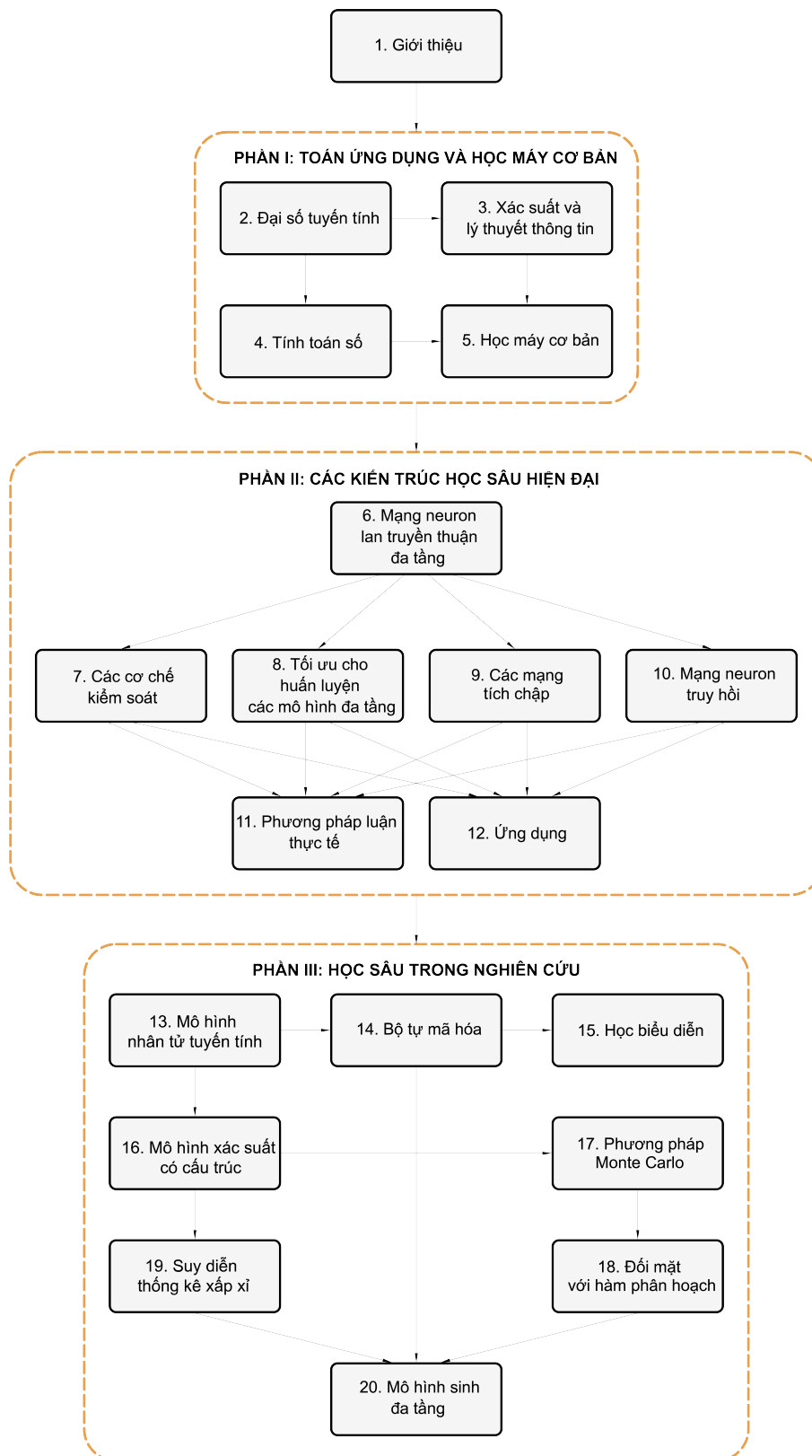
Hình 1.5: Biểu đồ cho thấy mối liên hệ giữa các thành phần khác nhau của một hệ thống AI trong mỗi hướng tiếp cận. Hình tô đậm là các thành phần có thể học được từ dữ liệu.

tìm hiểu về học máy, bao gồm những người đang bắt đầu sự nghiệp nghiên cứu trong lĩnh vực trí tuệ nhân tạo và học sâu. Thứ hai là các kỹ sư phần mềm, những người chưa có kiến thức nền tảng về học máy hay thống kê, nhưng muốn nhanh chóng tiếp thu kiến thức trong hai lĩnh vực này và bắt đầu sử dụng học sâu trong sản phẩm của mình. Học sâu đã chứng minh tính hữu dụng trong nhiều lĩnh vực phần mềm, bao gồm thị giác máy tính, xử lý âm thanh và tiếng nói, xử lý ngôn ngữ tự nhiên, robot, tin sinh học và hóa học, trò chơi điện tử, công cụ tìm kiếm, quảng cáo trực tuyến và tài chính.

Cuốn sách này được chia thành ba phần để giúp đỡ cho nhiều đối tượng đọc giả khác nhau một cách tốt nhất. Phần I giới thiệu các công cụ toán học cơ bản và những khái niệm học máy cơ bản. Phần II mô tả những thuật toán học sâu phổ biến và đã được phát triển một cách hoàn thiện nhất. Phần III mô tả những ý tưởng được suy đoán là quan trọng cho việc nghiên cứu học sâu trong tương lai.

Bạn có thể bỏ qua bất cứ phần nào trong sách nếu không quan tâm hoặc không phù hợp với kiến thức nền tảng của bạn. Ví dụ, đọc giả quen thuộc với đại số tuyến tính, xác suất, và khái niệm học máy cơ bản có thể bỏ qua phần I, trong khi đó những người chỉ muốn áp dụng học máy vào sản phẩm sẽ không cần quan tâm tới phần III. Để giúp bạn lựa chọn chương cần đọc, chúng tôi cung cấp một lưu đồ tổng quan thể hiện bố cục nội dung trong sách.

Chúng tôi tạm giả sử rằng tất cả các bạn đọc đều có hiểu biết cơ bản về khoa học máy tính như: lập trình, hiểu biết cơ bản về các vấn đề hiệu năng tính toán, lý thuyết về độ phức tạp của thuật toán, *phương pháp tính* (calculus) cơ bản và một số thuật ngữ trong lý thuyết đồ thị.



Hình 1.6: Tổng quan bố cục nội dung cuốn sách. Mũi tên từ chương này tới chương khác có ý nghĩa rằng cần phải đọc chương xuất phát của mũi tên mới có thể hiểu được chương mà mũi tên chỉ đến.